25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1910
METHODOLOGY OF AUTOMATIC MATHEMATICAL MODELING APPLIED TO A COMPACT INDIVIDUAL VEHICLE

Myagyh A. P. Miranda
Nestor Roqueiro
Universidade Federal de Santa Catarina – UFSC, Florianópolis, SC, Brasil
e-mails: myagyh.a.p.miranda@posgrad.ufsc.br, nestor.roqueiro@ufsc.br

Rodrigo De S. Vieira
Universidade Federal de Santa Catarina – UFSC, Florianópolis, SC, Brasil
e-mail: rodrigo.vieira@ufsc.br

Abstract. *It has always been a challenge to design systems and data, through mathematical models. Since ancient times, the human being seeks to mathematically describe experimental systems to help he understand them and thus to solve issues related to them. The objective of the work reported in this paper is to develop a methodology to automatically generate the mathematical modeling of dynamic systems, whether simple or advanced, applied to the case study, to the Individual Compact Vehicle (VIC). It was used Maplesoft's math-based modeling tool, MapleSim ™, built on the Maple symbolic computing system. Directly from the description of the block diagram, the analytical equations of the system are automatically generated and simplified with a powerful symbolic mechanism. The automatic generation of equations is used in a mathematical tool to analyze systems, and this design process drastically reduces the complexity of modeling. It is expected that the results generated by this methodology can receive applications on top of the models, not only on control projects but also on optimization, robustness analysis or any other type of analysis.*

Keywords: *Methodology, Dynamic systems, Mathematical models, Maplesim*

1. INTRODUCTION

Systems modeling is of utmost importance for the development of science and technology. It has always been a challenge to conceive, through mathematical models, the systems and the observed data. Since ancient times mankind seeks to mathematically describe experimental systems to help to understand them, and thus to solve questions related to them.

Mathematical modeling is involved in the study of dynamic systems, constituting a set of objects grouped by some interaction or interdependence. This is done in a way that there are relations of cause and effect in the phenomena that occur with the elements of that set; and that the response of its constituent objects varies in time (Monteiro 2006). Being most of the times of interest, the study of dynamic multi-domain system projects, whose conventional design differs from electronic circuits, mechanical systems, and fluid energy systems, are in part due to the need to integrate various types of energy behavior as part of the basic project (Seo et al. 2003).

Considering that the mathematical modeling of these dynamic systems is presented by a set of differential equations that can be obtained through study and manual calculation, it is used the physical laws that rule a particular system, such as, for example, Kirchhoff's laws of electrical systems, or Newton's laws of mechanical systems (Ogata 2014). Also, equations that represent the dynamics of the system accurately or at least quite acceptable. The aim of the work reported in this paper is to develop a methodology that will automatically generate the mathematical modeling of dynamic systems, whether simple or advanced.

It should be noted that a mathematical model is not unique to a given system, it can be described in different ways. Therefore, there may be several mathematical models, depending on the perspective that is analyzed. It is possible to improve the accuracy of a mathematical model by increasing its complexity, so in a complete system, hundreds of equations are included to describe it. With that in mind, it was used Maplesoft's math-based modeling tool, MapleSim ™, built on the Maple symbolic computing system, to assist in the automatic retrieval of the mathematical model, applied to a proposed case study, a Compact Individual Vehicle (CIV) with the intuition of performing some applications on the obtained model.

In general, when solving a new problem, it is desirable to first architect a simple model in order to acquire basic and general information to the solution. Subsequently, a more complex mathematical model can then be elaborated and used for a more detailed analysis.

1.1 Related Work

Despite there is not an automatic way to get the equations, there are some works using the software Maple in the development and use of models, such as Assaid (2011) which seeks to use it to determine the exact solution of the Poisson equation in a Schottky barrier junction by applying three different approaches. Mulero e Tian (2013) used the Maple™ in his work to develop an algorithm for deriving Differential Algebraic Equations (DAEs) and to generate new equations of States that are valid for the Hard-Sphere fluids (HS) or Hard-Disk (HD) that are common in the field of physics, chemistry, and chemical engineering.

Aazou, Assaid e Jadida (2009) the same tool was applied to study the exact analytical solution of equations characteristics of a solar cell, in your work, Rat (2011) presents dynamic analytical modeling of a parallel robot Triglide using numerical simulations through the Maple™ software and Adams. Seeking to speed up the processes of design and prototyping of hybrid electric vehicles, Dao e Friebe (2012) used the Maple™ and MapleSim™ because it's a multi-domain physical modeling tool for mathematical equations that govern the response of the model in a symbolic form.

Gachadoit e Renaud (2012) present better the advantages of this type of numerical tool and simulation, using as an example the modeling and design of an active suspension system. Wright e Soroka (2012) conducted modeling and simulation of piezoelectric linear stepper motor using the same tool and Ren (2016) to develop a simplified model in 1-D of a trial countertop of a machine gun system. But the work done by some authors or is forms of modeling method for optimization or application of its resources in a few specific cases, to generate the dynamic models or get his equations.

2. AUTOMATIC MODELING METHODOLOGY

In this section, it will be detailed the functions of each of the functional blocks presented in the flowchart of Figure 1's methodology, as well as the implementation decisions and the tools used in the development. The applied methodology focuses on obtaining the mathematical model using the previously mentioned computational tool, which assists in the automatic generation of the mathematical equations that rule the developed system response. This allows the system developer to perform better in terms of time and accuracy in the results generation. It is worth mentioning that the user needs technical and theoretical knowledge about the system to be developed so that the results generated during the process can be consistent with the domains to which the system belongs.

For the application of this methodology, it must be defined: either the object or conventional systems or the dynamic multi-domain system, so that they can be developed with Maple ™ and MapleSim ™. With this tool, nothing needs to be done or computed manually (with paper and pencil). Each stage of the process will be done by using the tool equipped in a computer, some additional toolboxes, templates and some available commands of execution.

It is desirable to have the model, that will be structuralized in the tool, developed in a CAD environment for the attainment of the properties of the selected problem constituent elements. When diversifying the tools performed in the model development, it is possible to approach the mathematical model to a real model against the precision of the obtained system applied data.

After making the block diagram model project with the physical components provided in the MapleSim™ library, parameterizing them and specifying their initial conditions as desired, the same must be performed, so that the model simulation verifies that its behavior is correct. Otherwise, the model can be modified until matching to specification. During this step (simulation), MapleSim™ automatically generates and simplifies the equations of the model and simulates them with the algebraic-differential solvers of the Maple ™. This is an important process because when generating the extraction of equations in Maple™, it will be obtained the exact set of equations since they have already been validated in the simulation which ensures that they are correct.

A physical system is usually represented in DAEs; therefore, this software uses differential equation solvers that employ advanced techniques to solve either Ordinary Differential Equations (ODEs), DAEs or Partial Differential Equations (PDEs). In addition, there is a toolbox that allows several conversions of these equations, being of interest in this paper to work with models in space states to carry out analyses in the obtained models.

In this way, when it is obtained the linear models, it is enough to use the conversion command of the system that generates the model in the system space state. Otherwise, there is a limitation where it is necessary to apply the linearization in the nonlinear model, to then carry out the transformation of the system space state. At this point, it causes the creation of uncontrollable and/or unobservable "virtual" states, depending on the complexity of the model. A reduction of the system is then applied to remove these states.

With the state space model done, it goes after to the analysis points and applications. It first evaluates the eigenvalues of the system, then it analyses their controllability and observability, in order to perform control applications to the model. In some cases, the user will obtain uncontrollable systems. At this moment, it is necessary to perform a canonical decomposition of the system to obtain the controllable and uncontrollable parts separating this part if there happens a control project in the system.

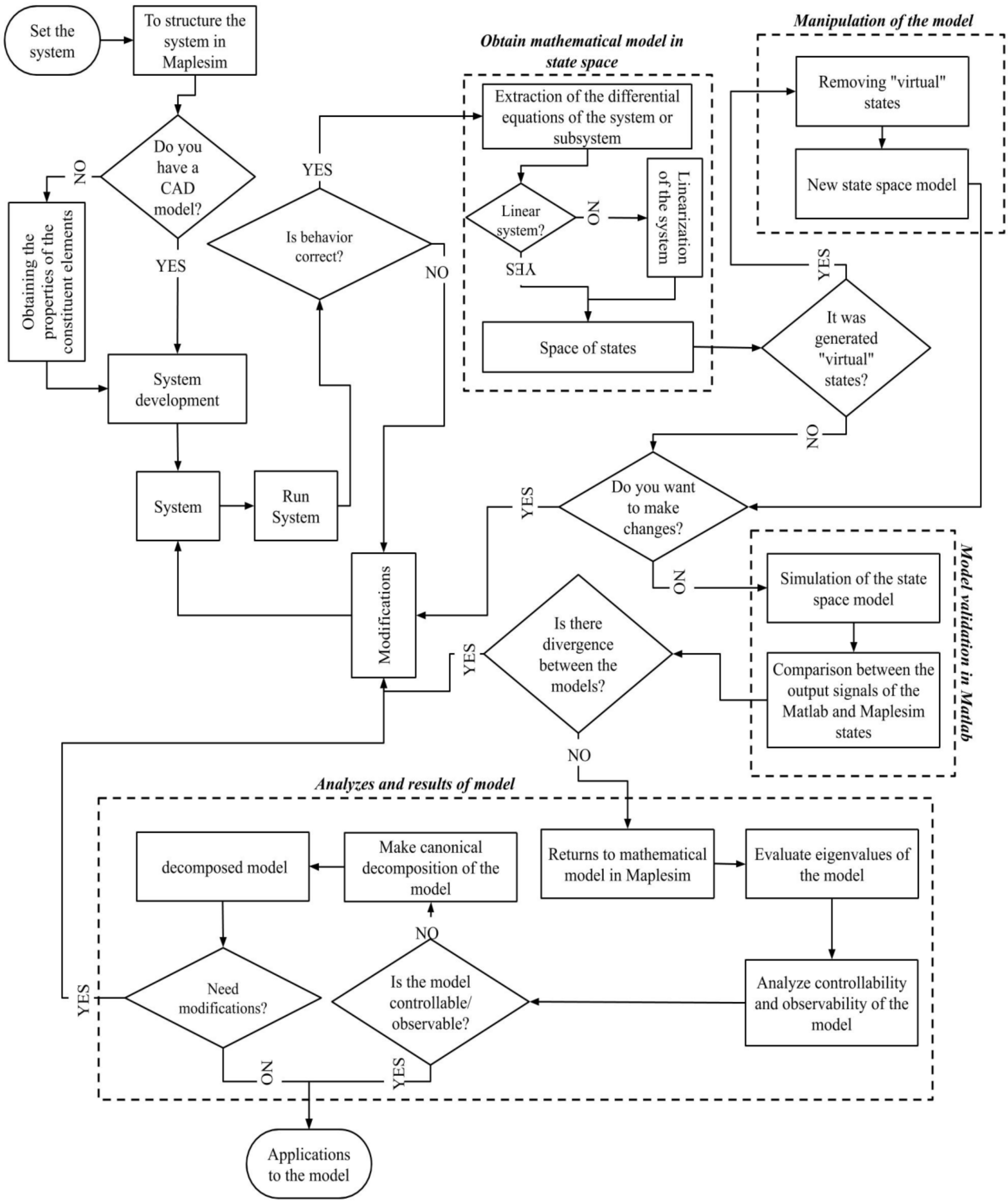


Figure 1: Flowchart of the methodology of obtaining mathematical models automated with the help of the tool Maple™ and MapleSim™.

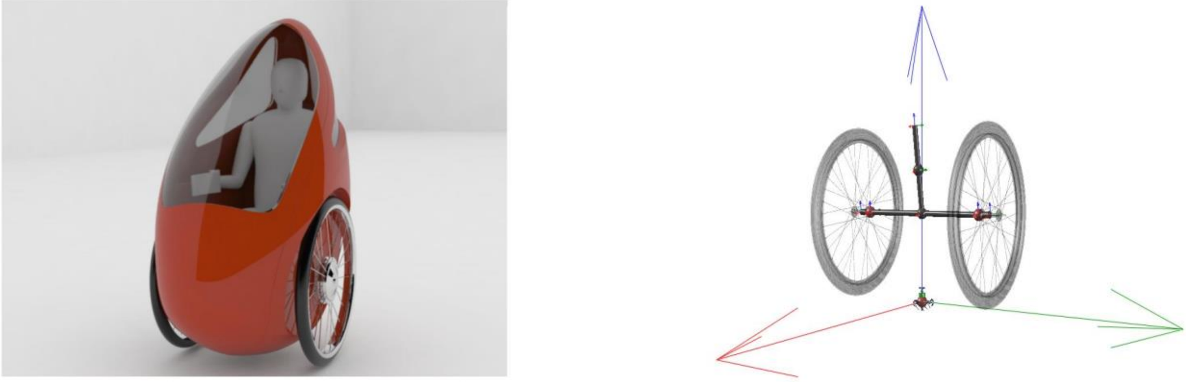
After obtaining and analyzing the model, the user can apply the desired control strategies, either in the mathematical model or the simulation model, as well as applying the already mentioned transformations in the system. For example, for functions of transference, discretization of the model or even through the returning of the model for distinguishing equations.

As a form of validation of the mathematical model obtained in the states space, the Matlab software was used, applying the model in state space and comparing the signals of exit of the desired system state with the signals gotten from the project developed in MapleSim™, which gave origin to the model. Thus, by validating the mathematical model of the system, in case of divergence, changes must be made to the model and its initial conditions and after replicating the steps.

The feasibility of applying this methodology was analyzed by using it to obtain a mathematical model of a Compact Individual Vehicle (CIV) and to perform applications on top of the given model. The obtained data will be presented and discussed in the next section.

3. COMPACT INDIVIDUAL VEHICLE (CIV)

In this session, based on the presented methodology, we will introduce the development of the VIC system. Figure 2 (a) shows the conceptual design of the vehicle. The system can be described as an inverted pendulum with two wheels, being the model based on work developed by Serafim, Roqueiro, and Braga (2018), as shown in Figure 2 (b).



(a) (b)

Figure 2: (a) conceptual design of the VIC. (b)Model-based on an inverted pendulum with two wheels

The vehicle model was implemented in MapleSim™ platform which is shown in Figure 3, which represents all components that structure the vehicle model and it is necessary to represent this way it for future extraction of equations taking into consideration all components. The properties of the components of this model were taken from the CAD model, in addition to applying the model initial conditions and restrictions as may be necessary to the execution an open loop simulation.

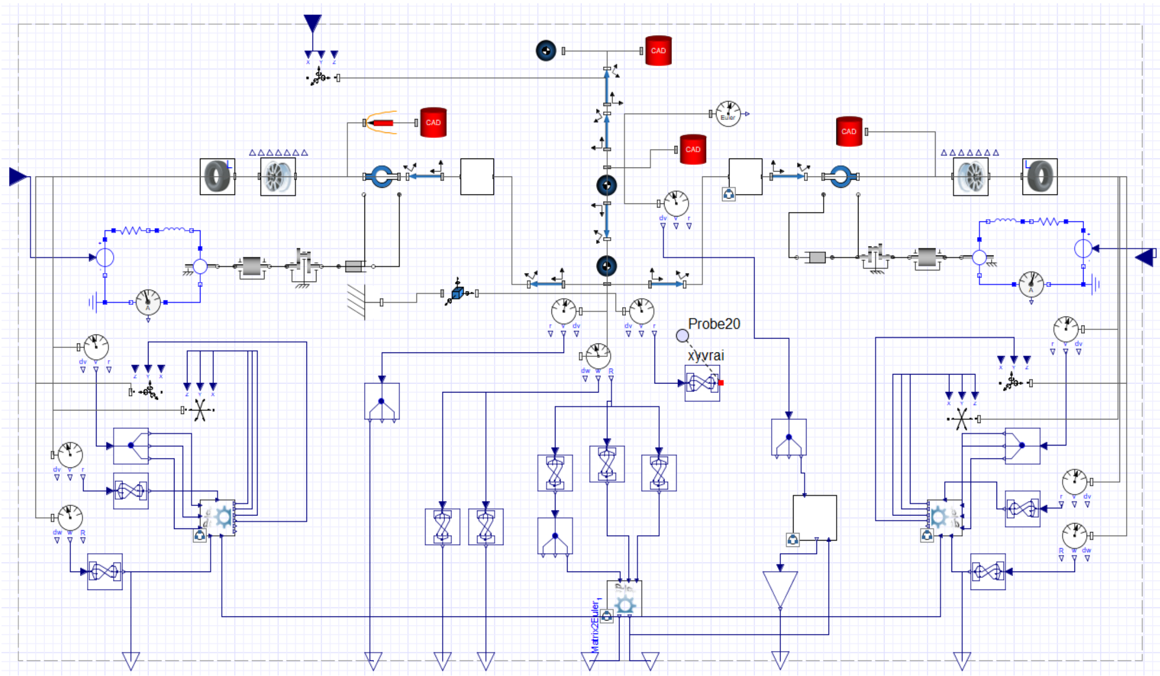


Figure 3: multibody Model VIC developed the MapleSim™.

With the implanted system, the generation of equations was done through the Maple™ from this given model. The system has 8 degrees of freedom and was modeled using 8 generalized coordinates; however, the equations are too complex to be shown in this article, even to be visually analyzed by the user, because of that, those equations are available through the QR code in Appendix A. Nevertheless, it is possible to identify that it is a non-linear system, and therefore, the system must be linearized in order to obtain the model in states space.

Two points were observed in the linearization process, which is important when analyzing the use of this process. The first one is related to the coordinate axis, which is fixed between the two wheels. In the MapleSim™, a restriction is applied that dislocates the axis for the desired point. However, when carrying through the process of linearization, the system does not recognize the axis of coordinates fixed between the two wheels of the vehicle, locating the coordinates as if they were in the origin. In this way, the coordinate was adjusted to the initial conditions making a displacement of 0,05 meters in relation to the coordinate z, returning the coordinating axis to the desired point without destabilizing the system.

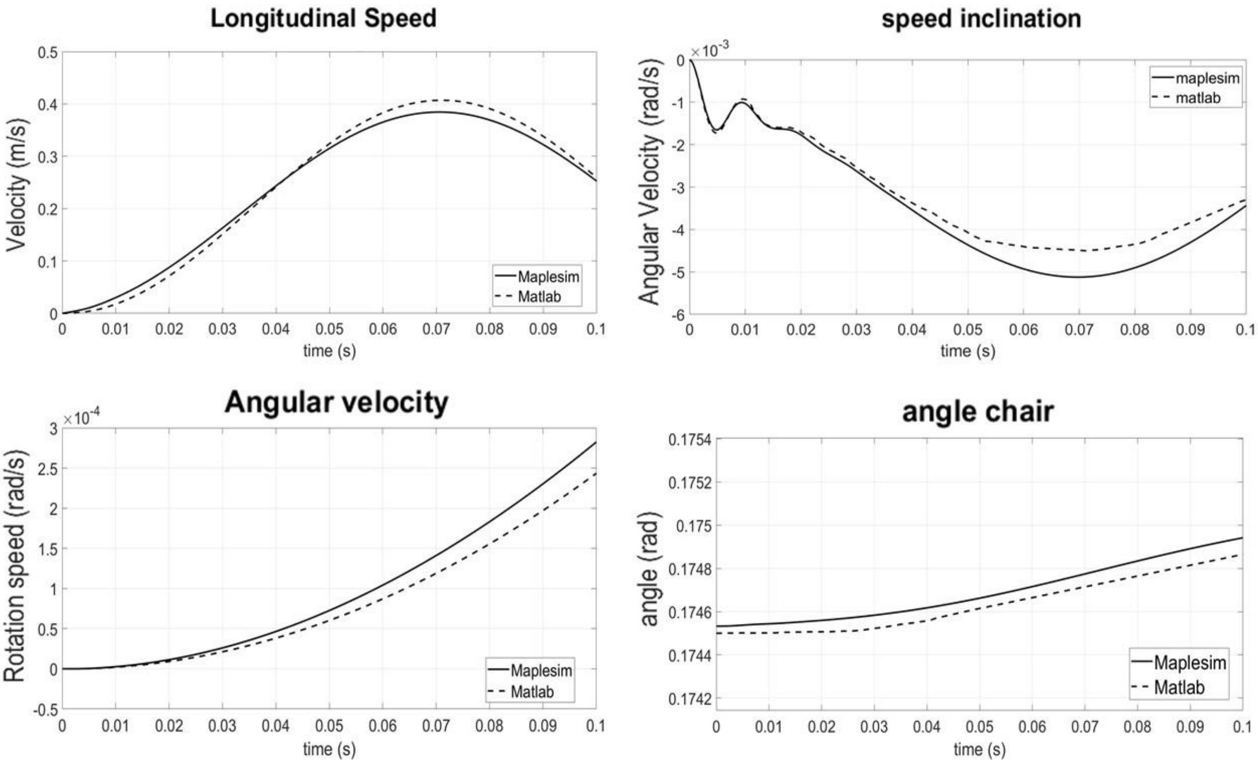


Figure 4: Graphics of the systems output signals in MapleSim™ open loop and Matlab.

Another observed point is regarding the generation of "virtual" states. After the linearization that resulted in the state space model with 22 states, it was applied the removal in these "virtual" states. This generated a model with 2 inputs, 4 outputs, and 17 states. Due to its complexity, the state arrays have not been incorporated in this article but are available for reference through the QR code (Appendix B), Which also contain the mathematical equations that govern the system response. As a form of validation, the obtained model was applied to the Matlab that generated output signals of the desired states. Thus, the signals from both open-loop (MapleSim™ and Matlab) systems were compared as it is illustrated in Figure 4

Because it is an open loop system, the simulations were performed in a short period of time, so it was possible to observe the behavior of the system. Since the linearization is an approximation around an operating point, it can only lead to the prediction of the behavior of the system in a neighborhood of this point. It makes it possible to observe that the signals have the same behavior, having a difference in relation to the non-linear system, MapleSim™, and the simulated linearized system through Matlab.

From this point on, the eigenvalues of the model were analyzed, in which the unstable states that characterize the inverted pendulum system were observed, as well as the system observability, yet the system was not controllable. Starting from the stage in which the canonical decomposition of the system is performed, at this stage the controllable and uncontrollable parts are separated. With this step it is possible to employ a controller to generate the stability in the system, the model of the controllable part can be seen in the QR code in Appendix B.

4. THE CIV CONTROLLER

The method used to design the controller's classic control theory. This is the project for allocation of poles for state space systems. The system control can be seen in Figure 5.

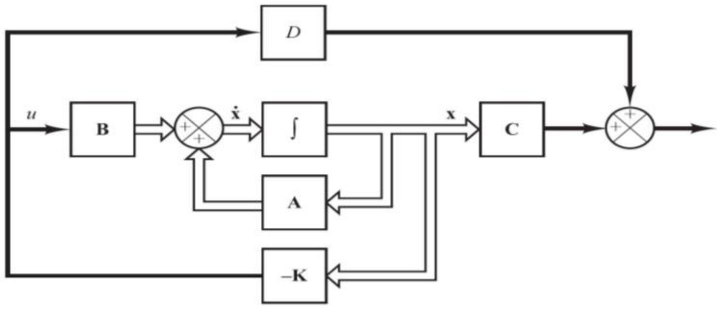


Figure 5: Closed loop control system

According to the structure presented in Figure 5 the project of the linear controller was carried based on the controllable model obtained in the canonic decomposition. It was used the technique of pole allocation, and was considered the following desirable poles in closed-loop:

$$p = [-50.13 - 50.1 - 50.02 - 50 - 49 \;\; -7 - 5 - 3.13 - 3.01 - 3] \tag{1}$$

The profits found were:

$$k = \begin{bmatrix} -83{,}06 & 26{,}66 & -5{,}21 & -5{,}46 & -251{,}63 & -31{,}92 & -23{,}83 & -23{,}85 & -0{,}016 & 0{,}269 \\ -30{,}40 & -26{,}49 & 5{,}70 & -4{,}60 & -224{,}14 & 33{,}22 & 23{,}83 & -23{,}87 & 0{,}275 & -0{,}275 \end{bmatrix} \tag{2}$$

4.1 Results through simulation

The control project previously seen was applied to the nonlinear model developed in MapleSim™. This paper presents the application of the controller supported in the model in continuous time, with the objective of verifying the dynamics of the system in closed-loop and the feasibility of the presented methodology.

The route that started at the marked point according to Figure 6 was simulated based on the minimum curve radius the vehicle performs. Therefore, the part indicated in Figure 6 shows the moment when the CIV has already reached stability, and when there are not considered inclinations or irregularities in the terrain.

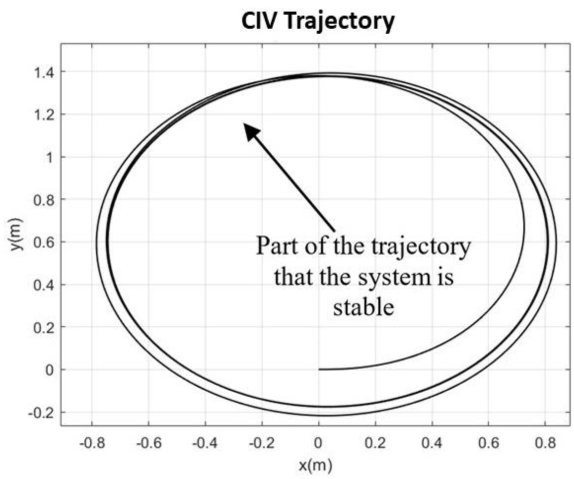


Figure 6: Trajectory simulation

The general results of the simulations are presented in Figure 7 and Figure 8, showing the viability of the methodology to perform applications. As in the case here presented; a control application on the model in the desired states to be controlled, the set of graphs are:

1º - Tilt angle in the longitudinal direction of the vehicle;

2º - Tilt speed of the vehicle;

3º - Motor voltage;

4º - Longitudinal speed;

5º - Angular speed.

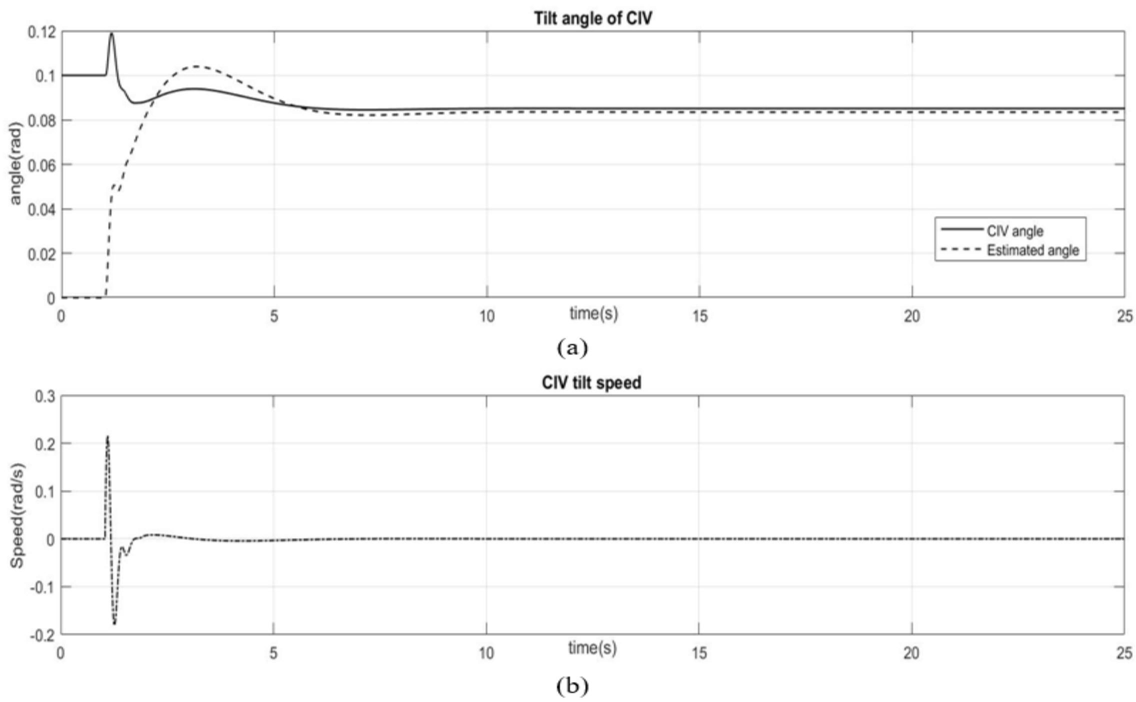
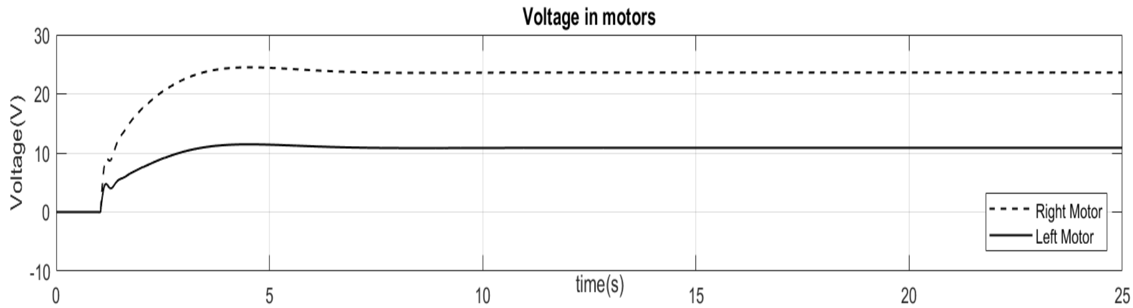


Figure 7: control result on (a) the angle and (b) the tilt speed CIV.

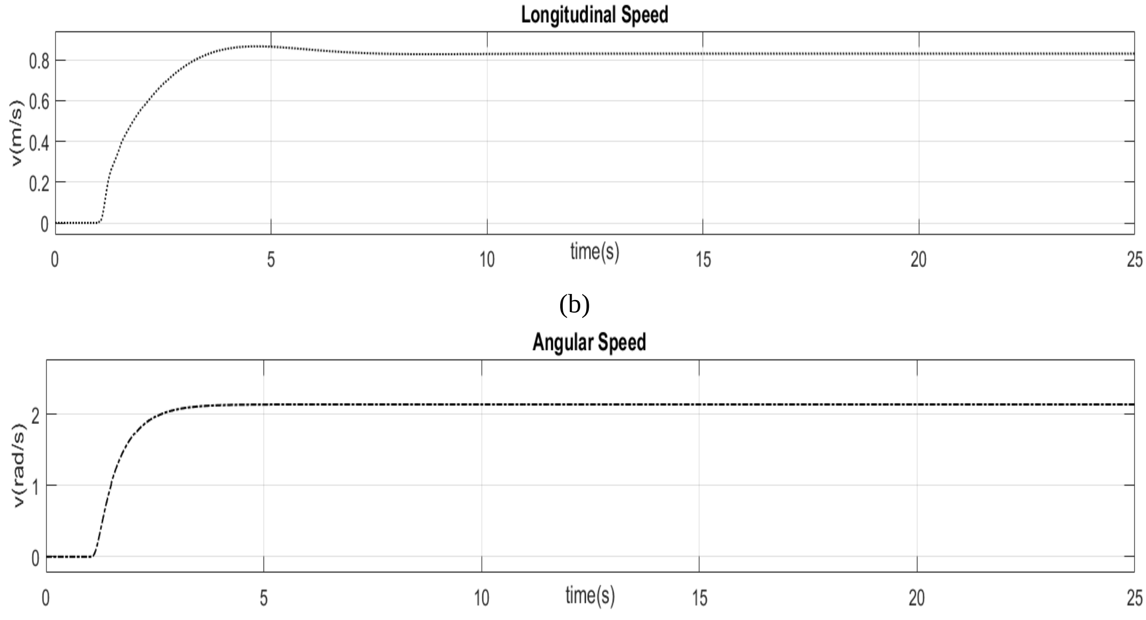
As can be seen, the control project presented a satisfactory result. Looking at Figure 7(a), the dashed curve represents the estimated tilt angle of the vehicle started at 0 rad, and the curve in a continuous line of the simulated vehicle angle starts at 0.1 rad (this fact is due to initial conditions imposed on the developed model). Despite the initial condition, both angles stabilize just after the initial 5 seconds at approximately 0.085 rad.

Another fact that certifies the control of vehicle angle can be seen in Figure 7(b), which shows that the period of oscillations present at the tilt speed ceases at the same time as the angle is stabilized.

On Figure 8(a), the maximum tensions applied to the step motors were of 24 V on the motor 1 and of 10.59 V on the motor 2, enabling the CIV to perform the path with minimum radius curve. The maximum longitudinal speed reached, in this case, was of 0.8 m as can be seen in Figure 8 (b). In Figure 8 (c), the angular speed, that is, the rotational speed of the vehicle in performing curves, was of 2.14 rad/s.



(a)



(c)

Figure 8: Simulation results: (a) motor voltage, (b) longitudinal velocity and (c) angular velocity.

5. CONCLUSIONS

In this paper, it was presented the development and application of a methodology for the automatic generation of mathematical models from a block diagram applying the benefits offered by the computational tools here mentioned. The developed methodology was applied to the compact individual vehicle allowing a high acceleration of the development process, and no manual computation of equations was necessary, which is a unique characteristic.

The proposed methodology can generate a dynamic model, admitting applications such as the control ones. The development of the control project used in the system obtained responses that were consistent with the expected results. This generated a contribution to the study of two-wheel dynamic mobile manipulation platforms, to future research for engineers, to the development of sophisticated control strategies, as well as to the optimization, robustness analysis, etc.

6. ACKNOWLEDGEMENTS

This work received financial support from the Coordination of Improvement of Higher Level Personnel (CAPES).

7. REFERENCES

Aazou, Safae, El Mahdi Assaid, and El Jadida. 2009. “Modelling Real Photovoltaic Solar Cell Using Maple.” In *2009 International Conference on Microelectronics*, , 394–97.

Assaid, El Mahdi, Safae Aazou, Asmaa Ibral, and El Jadida. 2011. “Finite Difference Numerical Solution of Poisson Equation in a Schottky Barrier Diode Using Maple.” In *2011 Faible Tension Faible Consommation (FTFC)*, , 123–26.

Dao, Thanh-son, and Johannes Friebe. 2012. “Symbolic Techniques for Model-Based Design of Hybrid Electric Vehicles.” In *2012 2nd International Electric Drives Production Conference (EDPC)*, , 1–10.

Gachadoit, N, and R Renaud. 2012. “Modeling and Design of an Active Suspension System With Maple and Maplesim.” In *2012 9th France-Japan & 7th Europe-Asia Congress on Mechatronics (MECATRONICS) / 13th Int’l Workshop on Research and Education in Mechatronics (REM)*, , 425–32.

Monteiro, L. H. A. 2006. *Sistemas Dinâmicos*. São Paulo: Editora Livraria da Física.

Mulero, Ángel, and Jianxiang Tian. 2013. “Hard-Sphere Equations of State and Maple.” *Computing in Science & Engineering*: 56–62.

Ogata, Katsuhiko. 2014. *Engenharia de Controle Moderno*. 5º. Prentice Hall.

Rat, Nadia Ramona, Mircea Neagoe, Dorin Diaconescu, and Sergiu Dan Stan. 2011. "Dynamic Analysis of a Triglide Parallel Robot." In *2011 4th International Conference on Human System Interactions, HSI 2011*, , 245–49.

Ren, Haifeng. 2016. "Modeling and Simulating in MapleSim for a Bench Experimental System of Machine Gun." In *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAl)*, , 808–12.

Seo, Kisung et al. 2003. "Toward a Unified and Automated Design Methodology for Multi-Domain Dynamic Systems Using Bond Graphs and Genetic Programming." *Mechatronics* 13(8-9 SPEC.): 851–85.

Serafim, João Marcel, Nestor Roqueiro, and Rodrigo A. M. Braga. 2018. "Modelagam, Análise e Controle de Um Protótipo de Cadeira de Rodas Auto-Equilibrista." *13th IEEE International Conference on Industry Applications (INDUSCON)*: 9.

Wright, Derek, and Orysia Soroka. 2012. "Modeling and Simulation of a Linear Piezoelectric Stepper Motor in MapleSim." In *2012 IEEE International Ultrasonics Symposium*, , 2746–49.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.

APPENDIX A

APPENDIX B